

Sistem Pembayaran Berbasis Web di Kantin Universitas Pertamina

Arina Meitri¹, Daniel Fetrick¹, Disa Maya¹, Muhammad Qory¹, Nurul Isnaeni¹, Nanda Ruswandi^{1*}, Adji Candra Kurniawan¹, A.A.N. Perwira Redi²

¹Logistics Engineering, Universitas Pertamina, Jl Teuku Nyak Arief, South Jakarta 12220, Indonesia

²Department of Industrial Engineering, Binus Graduate Program, Binus University, Jl. Kebon Jeruk Raya No. 27, Kebon Jeruk, West Jakarta 11530, Indonesia

Abstrak

Perkembangan teknologi dapat mempermudah manusia melakukan aktivitasnya, salah satu pemanfaatan teknologi tersebut adalah membantu dalam proses pembayaran. Artikel ini menyajikan tahapan dalam pembangunan sistem pembayaran berbasis *web* di kantin Universitas Pertamina yang nantinya akan berfungsi untuk memudahkan dalam proses pemesanan makanan, memudahkan pemilik untuk mengecek hasil penjualan mereka dan juga meminimalkan antrian dalam pemesanan sehingga mampu memproses pesanan dengan cepat dan tepat. Dengan menggunakan metode observasi, wawancara kepada pemilik kantin di Universitas Pertamina dan menggunakan beberapa studi literatur diharapkan dapat memenuhi keinginan dan harapan untuk mempermudah pemesanan dan pembayaran di kantin Universitas Pertamina ini. Bahasa pemrograman untuk aplikasi sistem pembayaran di Universitas Pertamina menggunakan MySQL untuk basis data, phpMyAdmin untuk mengelola database MySQL, javascript digunakan sebagai bahasa pemrograman, dan XAMPP versi 7.4.4. XAMPP merupakan sebuah paket program yang memuat Apache *web server* di dalamnya. Paket program ini dapat digunakan pada jaringan komputer lokal sehingga dapat digunakan untuk melayani tampilan halaman *web* dan mampu membangun sebuah *prototype* aplikasi sistem pembayaran. Hal ini sesuai dengan kebutuhan yang dibutuhkan oleh para pemilik kantin seperti data *supplier*, jumlah *customer*, *products*, jumlah transaksi pada hari ini dan juga *reports* untuk melihat rekap transaksi.

Kata kunci: Sistem pembayaran *web*; *Cafeteria point of sales*; Sistem informasi

Abstract

Technological developments can make it easier for humans to carry out their activities, one of the uses of this technology is to assist in the payment process. This article presents the stages in the development of a web-based payment system at the Pertamina University canteen which will later function to facilitate the food ordering process, make it easier for owners to check their sales results, and also minimize queues in orders so they can process orders quickly and precisely. By using the method of observation, interviews with the owner of the canteen at Pertamina University and using several literature studies are expected to fulfill the wishes and expectations to facilitate ordering and payment at the Pertamina University canteen. The programming language for payment system applications at Pertamina University uses MySQL for databases, phpMyAdmin to manage MySQL databases, javascript is used as a programming language, and XAMPP version 7.4.4. XAMPP is a program package that contains the Apache webserver inside. This program package can be used on a local computer network so that it can be used to serve web page displays and is able to build a payment system application prototype. This is in accordance with the needs required

*Corresponding author

Alamat email: nanda.ruswandi@gmail.com

by canteen owners such as supplier data, a number of customers, products, number of transactions today, and also reports to see transaction recap.

Keywords: *Web payment system; Cafeteria point of sales; Information system*

Pendahuluan

Universitas Pertamina merupakan salah satu perguruan tinggi swasta yang terletak di Jakarta yang berdiri pada tahun 2016, didirikan berdasarkan Surat Keputusan (SK) Menteri Riset, Teknologi, dan Pendidikan Tinggi No. 60/KPT/I/2016. Berdirinya Universitas Pertamina merupakan wujud kontribusi PT. Pertamina (Persero) kepada masyarakat melalui penyelenggaraan pendidikan tinggi, Universitas Pertamina didirikan dengan semangat menjadi perguruan tinggi berkelas dunia di bidang bisnis dan teknologi energi. Universitas Pertamina saat ini memiliki 6 (enam) Fakultas dan 15 Program Studi strata satu [1]. Kampus ini memiliki berbagai macam fasilitas penunjang untuk kegiatan mahasiswa diantaranya laboratorium, ruang kelas dan kantin.

Kantin merupakan fasilitas yang sering dikunjungi mahasiswa untuk makan, minum dan berbelanja hal-hal keperluan lain. Aktivitas ini tidak lepas dari proses transaksi jual beli dengan sistem pembayaran tunai. Hal tersebut tentunya tidak efisien karena harus menyediakan uang kembalian. Sistem tersebut juga menimbulkan ketidaknyamanan dalam bertransaksi yaitu akan terjadi antrian dalam bertransaksi [2]. Apalagi jika pelanggan meminta bukti transaksi. Bukti transaksi harus dapat tersimpan dan tertulis secara baik dan mudah untuk diakses agar tidak terjadi kerugian. Tidak salah bila kesalahan transaksi tersebut akibat Human Error atau sistem bukti transaksi yang masih menggunakan bukti fisik yakni nota/struk [3]. Proses pembayaran tidak dilakukan secara langsung oleh sistem sehingga terkadang mengalami kesulitan dalam penanganan antrian dan laporan keuangan yang kurang memadai [4]. Hal ini dapat dilihat dari meningkatnya jumlah mahasiswa yang ada di Universitas Pertamina. Pada tahun 2016 jumlah mahasiswa Universitas Pertamina adalah 1235. Jumlah ini mengalami peningkatan signifikan yaitu sebesar kurang lebih 3 kali lipat sehingga menghasilkan jumlah mahasiswa sebesar 4967 pada tahun 2019 [5]. Peningkatan jumlah mahasiswa tidak diimbangi dengan meningkatnya fasilitas warung di kantin. Berdasarkan pengamatan peneliti, jumlah warung berkisar 8 (delapan) sampai 11 buah dari tahun 2016 sampai 2019. Hal ini berarti jika diasumsikan distribusi merata mahasiswa dalam mengunjungi setiap warung, maka rata-rata antrian di setiap warung dengan hanya 40% dari total mahasiswa sudah mencapai 15 orang per warung setiap 5 menit. Angka tersebut dinilai sudah cukup menggambarkan potensi antrian yang ada di kantin Universitas Pertamina.

Beberapa kantin di Universitas Pertamina masih menerapkan sistem pembayaran tunai dan ada beberapa kantin sudah menerapkan sistem pembayaran non-tunai. Semakin mudahnya akses Internet broadband di Indonesia, serta teknologi ponsel pintar yang mendukung aktivitas sehari-hari, mengakibatkan perubahan gaya hidup masyarakat dalam berbagai hal [6], seperti halnya sistem barcode yang bisa discan atau GoPay, OVO, Atau seperti penelitian sebelumnya yaitu sistem perancangan pembayaran SPP berbasis *web* dan *SMS Gateway* [7] yang sistem pembayaran menggunakan informasi SMS, serta ada sistem pembayaran yang bernama MOKA, yang merupakan sistem pembayaran yang dilengkapi dengan akses ke data-data penjualan dari sebuah toko/kios dengan sistem pembayaran seperti ini memudahkan kita baik sebagai penjual maupun pembeli bertransaksi. Dalam hal ini akan dirancang sistem pembayaran yang mampu menyimpan data-data transaksi hasil penjualan dari sebuah kios di kantin Universitas Pertamina.

Sistem yang dirancang untuk sistem pembayaran kantin Universitas Pertamina ini termasuk jenis sistem Transaction Processing System (TPS). Sistem yang dibuat merupakan sistem yang berbasis *web*, pembangunan sistem berbasis *web* memiliki beberapa keunggulan antara lain mudah untuk dikembangkan karena bahasa pemrograman yang digunakan sudah sangat umum, mudah diakses oleh setiap pengguna, dan fleksibel karena dapat menyesuaikan dengan berbagai *devices*. Sistem pembayaran yang akan memiliki *database* yang bisa digunakan oleh masing-masing kios untuk menyimpan data-data tentang *user*, *admin* dan produk-produk yang dijual pada kios penjual itu sendiri. Dengan adanya rancangan sistem ini, diharapkan dapat meningkatkan efisiensi dalam mencatat data transaksi pada setiap kios yang ada di kantin. Untuk kemudahan pembahasan sistem pembayaran berbasis *web* di kantin Universitas Pertamina juga disebut sebagai sistem *e-kantin*.

Studi Pustaka

Menurut Romney dan Steinbart sistem adalah kumpulan dari dua atau lebih komponen yang saling bekerja dan saling berhubungan untuk mencapai tujuan tertentu. Beberapa ciri-ciri sistem yaitu sistem mempunyai komponen-komponen yang membentuknya, komponen sistem harus terintegrasi, sistem mempunyai batasan, sistem mempunyai tujuan yang jelas, sistem mempunyai lingkungan, dan sistem mempunyai *input*, proses, dan *output*. Sistem informasi merupakan sistem yang terdiri dari kumpulan teknologi, fasilitas, prosedur, pengendali, dan media [8]. Sistem informasi menggunakan teknologi yang terdiri dari perangkat keras dan perangkat lunak. Sistem informasi terdiri dari empat jenis yaitu *Transaction Processing System* (TPS), *Management Information System* (MIS), *Decision Support System* (DSS), dan *Enterprise Resource Planning System* (ERP).

Transaction Processing System (TPS) adalah sistem informasi yang terkomputerisasi yang dikembangkan untuk memproses data-data dalam jumlah besar untuk transaksi bisnis rutin, seperti inventarisasi. TPS berkembang dari sistem informasi manual untuk sistem proses data dengan bantuan mesin menjadi sistem proses data elektronik data dari hasil transaksi bisnis, seperti penjualan, pembelian dan perubahan data persediaan (*inventory*). Sistem ini menghasilkan berbagai informasi produk untuk penggunaan internal maupun eksternal. Biasanya sistem ini digunakan langsung oleh *front liner* perusahaan seperti, kasir atau resepsionis. Data yang diolah pada sistem informasi model ini adalah data yang terotomatisasi atau semi otomatis pada aktivitas perusahaan level rendah dan transaksi dasar.

Metode *Waterfall* atau metode air terjun adalah salah satu metode pengembangan sistem. Menurut Pressman, model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah "*Linear Sequential Model*". Model ini sering disebut juga dengan "*classic life cycle*" atau metode *waterfall*. Model ini melakukan pendekatan secara sistematis dan berurutan, sehingga tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan [9]. Metode *waterfall* dianggap lebih cocok sebagai pendekatan yang digunakan untuk proyek pembuatan sistem baru dan juga pengembangan *software* dengan tingkat risiko yang kecil serta waktu pengembangan yang cukup lama. Berikut ini adalah tahapan dari metode *waterfall* yaitu: *Requirements Analysis and Definition*, *System and Design Software*, *Implementation*, *Integration and Testing*, dan *Operation and Maintenance*.

Metode Penelitian

Metode pengumpulan data yang diterapkan dalam perancangan sistem pembayaran berbasis *web* di Kantin Universitas Pertamina yaitu : observasi, wawancara, dan *benchmark study*. Observasi dilakukan langsung di kantin Universitas Pertamina. Wawancara dilakukan kepada beberapa penjual atau pemilik toko di kantin Universitas Pertamina dan dosen mata kuliah Sistem Informasi yang *expert* di bidang sistem informasi. *Benchmark study* dilakukan dengan mencari referensi teori dan contoh-contoh aplikasi yang relevan dengan permasalahan yang ditemukan.

Selain metode pengumpulan data, peneliti juga menerapkan metode pengembangan sistem informasi. Metode yang digunakan pada pengembangan sistem pembayaran berbasis *web* yang akan diterapkan di Kantin Universitas Pertamina adalah menggunakan metode *waterfall*. Metode *waterfall* dipilih dikarenakan melakukan fase secara berurutan dan tidak dapat melakukan pengerjaan fase secara bersamaan, sehingga meminimasi terjadinya kesalahan yang mungkin terjadi. Tahapan dalam melakukan pengembangan sistem pada metode *waterfall* [10] antara lain: *Requirements Analysis and Definition*, *System and Design Software*, *Implementation*, *Integration and Testing*, dan *Operation and Maintenance*.

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem [11]. Tahapan pertama dalam melakukan pengembangan sistem *e-kantin* yaitu dengan mengumpulkan informasi atau data yang diperlukan. Pengumpulan data dilakukan untuk mespesifikasikan adanya kebutuhan perancangan sistem agar nantinya sistem pembayaran kantin berbasis *web* tersebut dapat dipahami dan sesuai dengan yang dibutuhkan *user*. Informasi atau data tersebut dapat diperoleh dari hasil wawancara dengan beberapa pihak kantin, melakukan *survey*, ataupun diskusi.

Tahapan perancangan sistem selanjutnya adalah melakukan fokus pada desain pembuatan sistem pembayaran berbasis *web* termasuk pembuatan struktur data pada sistem. Pada tahap ini dilakukan translasi kebutuhan perancangan sistem dari tahap analisis kebutuhan menuju representasi desain agar nantinya dapat diimplementasikan menjadi suatu sistem yang utuh. Desain suatu sistem menggunakan permodelan basis data yang dibentuk secara prosedural dengan menggunakan ERD (*Entity Relationship Diagram*), selain itu desain sistem juga digambarkan pada *use case diagram*, *activity diagram*, dan *sequence diagram*. Tujuan tahapan ini adalah untuk memberikan gambaran lengkap tentang apa yang nantinya harus dikerjakan dan bagaimana tampilan dari sebuah sistem yang diinginkan *user*.

Pada tahap selanjutnya, sistem yang telah melewati proses desain akan direalisasikan sebagai serangkaian program atau unit program, serta desain yang dibuat direpresentasikan dalam bahasa pemrograman berupa kode–kode program yang masih terbagi menjadi modul–modul yang akan diintegrasikan menjadi sistem yang lengkap. Pada kasus ini pembuatan bahasa pemrograman menggunakan pemrograman MySQL untuk basis data, phpMyAdmin untuk mengelola database MySQL, javascript digunakan sebagai bahasa pemrograman, dan XAMPP versi 7.4.4 yang digunakan sebagai pembuatan *web server* secara lokal yang dapat melayani tampilan halaman *web* yang dinamis, selain itu XAMPP sebagai *software package* yang berisikan server lokal sehingga dapat digunakan untuk menguji kinerja fitur tanpa harus ada koneksi internet. Pengimplementasian ini melibatkan adanya verifikasi agar dapat melakukan penyesuaian antara sistem yang dibuat apakah sudah memenuhi spesifikasinya.

Pada tahap berikutnya dilakukan penggabungan modul-modul yang tadinya dipecah, kemudian dilakukan *testing* yang bertujuan untuk mengetahui apakah desain sistem yang dibuat sudah sesuai dengan apa yang diinginkan, dan apakah masih ada *error* atau tidak, setelah lulus tahap pengujian sistem dapat dikirimkan ke *customer*.

Pada tahap akhir ini sistem yang dibuat sudah dapat dijalankan atau dioperasikan untuk setiap penggunaannya, jika dalam kasus ini sistem sudah dapat digunakan oleh penjual di Kantin Universitas Pertamina. Pada perbaikan sistem dapat melakukan pemeriksaan secara periodik terhadap data pada sistem pembayaran.

Hasil dan Pembahasan

Dalam bagian hasil dan pembahasa terdapat tiga tahapan pembuatan sistem yang dibahas yaitu tahapan desain sistem, rancangan basis data, dan desain antarmuka sistem. Desain sistem digunakan untuk merealisasikan kebutuhan spesifikasi sistem ke dalam bentuk arsitektur sistem, berikut langkah dalam memenuhi desain sistem pembayaran Kantin Universitas Pertamina berbasis *web*.

Tahap ini membantu dalam menspesifikasikan kebutuhan *hardware* dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan [12]. Pertama-tama kebutuhan fungsional dari sistem ini didaftarkan seperti yang diperlihatkan pada Tabel 1. *Functional requirement* merupakan kebutuhan yang berisi proses apa saja atau layanan yang nantinya dapat disediakan oleh sistem.

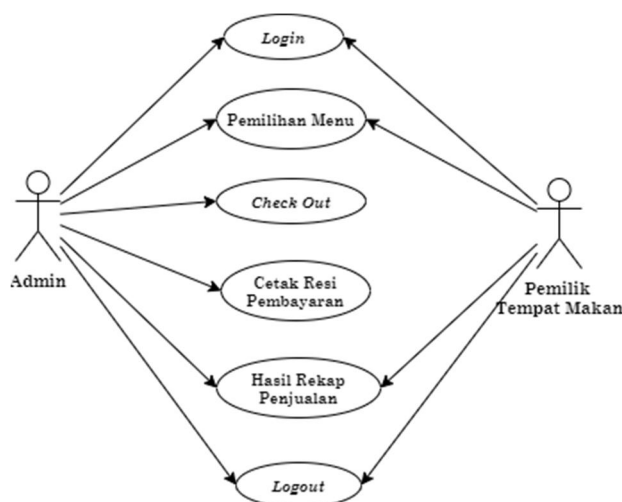
Tabel 1. Kebutuhan fungsional perancangan sistem *e-kantin* Universitas Pertamina

Kebutuhan	Penjelasan
<i>Login</i> pengguna/ <i>admin</i>	Sistem dapat memvalidasi pengguna saat pertama kali menggunakan sistem pembayaran
Fitur melihat menu	Admin dapat melihat <i>list</i> menu yang tersedia pada warung atau tempat makan.
Fitur pilih menu	Admin dapat melakukan pemilihan terhadap pilihan menu makan
Fitur pembayaran	Terdapat total pembayaran dan <i>admin</i> dapat mencetak bukti transaksi namun hal ini opsional
Fitur historis	Sistem transaksi digunakan untuk melihat data penjualan pada hari tersebut dan dapat digunakan untuk melihat data <i>historis</i>
Fitur <i>Log out</i>	Admin dan pemilik dapat melakukan opsi keluar atau <i>log out</i> dari sistem ini

Setelah kebutuhan fungsional, kemudian terdapat kebutuhan non-fungsional pada perancangan sistem *e-kantin* Universitas Pertamina yang dikategorikan dalam *performance*, *safety*, *security*, dan *software quality*. Sistem pembayaran kantin berbasis web di Universitas Pertamina memiliki *performance requirements* berupa suatu transaksi di dalam sistem *e-kantin* dapat dihandle pada satu waktu, serta setiap penggunaan sistem untuk transaksi memiliki *processing time* dan *response time*. Selain itu sistem pembayaran dibuat untuk meningkatkan sistem layanan transaksi lebih efektif terutama pada tahap pendaftaran, login transaksi, maupun penggunaan fitur yang ada dalam sistem. Terkait *safety* dan *security* sistem *e-kantin* Universitas Pertamina ini telah dilengkapi perintah login disertai dengan username dan password. Password yang telah dibuat oleh pengguna bersifat rahasia dan pribadi. *Security requirement* adalah keamanan dari penggunaan sistem yang dibuat. Pada sistem pembayaran berbasis *web* di kantin Universitas Pertamina memiliki akses akun yang berbeda setiap pengguna, serta sistem *e-kantin* dapat merekam setiap transaksi yang terjadi, maka penyelewengan

penggunaan sistem akan minim terjadi. Berikutnya terkait *software quality attributes* yang dibagi menjadi *reliability*, *portability* dan *usability*. Terkait *reliability requirement*, Sistem diharapkan dapat digunakan oleh pengguna selama 24 jam dalam sehari, asalkan selalu terkoneksi dengan *internet*. Sistem juga perlu bersifat *portable*, dikarenakan pengguna dapat melakukan akses transaksi dimana saja. Selain itu, sistem perlu memiliki instruksi yang memandu pengguna baru untuk melakukan pendaftaran, apabila pengguna baru salah input.

Setelah kebutuhan fungsional dan non-fungsional sistem didaftarkan, ilustrasi antara kebutuhan fungsional dan pengguna dijabarkan dalam *use case diagram and specification*. *Use case* menggambarkan *external view* dari sistem yang akan dibuat modelnya [13]. Setelah menjabarkan keseluruhan kebutuhan fungsional dan non fungsional pada sistem. Gambar 1 menampilkan *use case diagram* pada sistem *e-kantin Universitas Pertamina*.

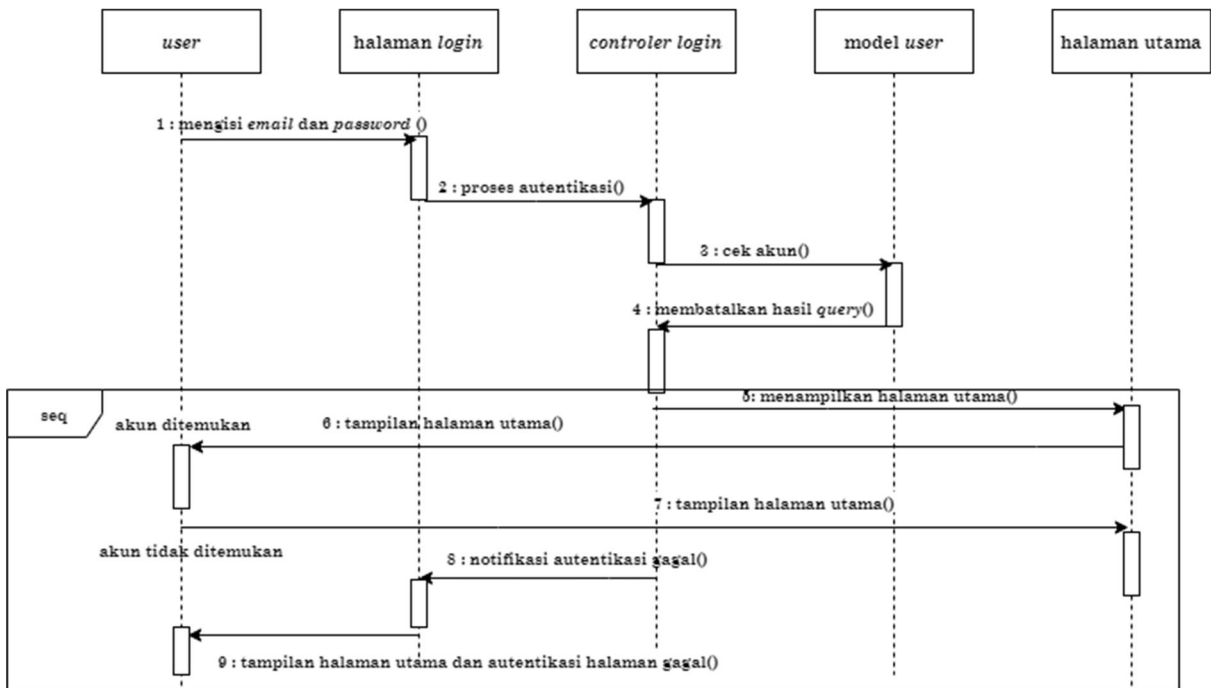


Gambar 1. *Use case diagram* sistem pembayaran berbasis web

Gambar 1 menjelaskan bahwa terdapat 2 aktor di dalam sistem pembayaran pada Kantin Universitas Pertamina yaitu admin dan pemilik tempat makan, kemudian terdapat beberapa *use case* utama diantaranya *login*, pilihan menu, *check out*, cetak resi pembayaran, hasil rekap penjualan, dan *logout*. Langkah berikutnya adalah *use case diagram* yang dibuat sebelumnya akan dijabarkan pada *use case spesifikasi* agar sistem yang diinginkan lebih jelas dan spesifik, daftar tabel penjabaran tiap *use case* pada *use case diagram* dapat dilihat pada Lampiran A. Kemudian, masing-masing tabel *use case spesifikasi* perlu dibuatkan *sequence diagram* yang menggambarkan urutan pada alur skenario yang terjadi pada sistem. Daftar *sequence diagram* yang ada pada sistem ini dapat dilihat pada Lampiran B. Tabel 2 dan Gambar 2 memberikan contoh terkait hubungan antara *use case spesifikasi* dan *sequence diagram*.

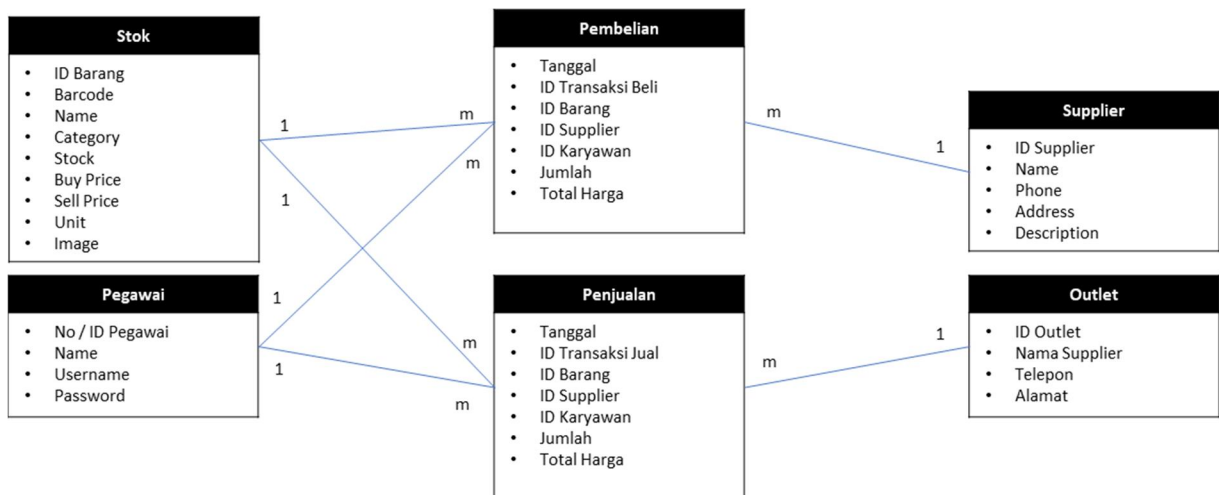
Tabel 2. RF-5: Hasil rekap penjualan

Nama Use Case	Hasil rekap penjualan	
Aktor	Pemilik tempat makan, kasir	
Entry Condition	Aktor membuka halaman Hasil Rekap Penjualan	
Scenario	Aktor	Sistem
	1. Klik tombol rekap penjualan 2. Klik hasil rekap sesuai kebutuhan (hari ini, 1 minggu terakhir, 1 bulan terakhir, 1 tahun terakhir)	1. Sistem menampilkan hasil rekap sesuai yang aktor inginkan
Exit Condition	Aktor mengecek hasil penjualan	






Gambar 2. Sequences diagram halaman rekap hasil penjualan

Setelah desain sistem dibuat, berikutnya adalah melakukan perancangan terkait skema basis data yang digunakan. Dalam sistem ini basis data yang digunakan berdasarkan skema *relational database* seperti yang diilustrasikan pada Gambar 3. Desain basis data merupakan proses perancangan dari sebuah *database* yang digunakan untuk mengimplementasikan sistem dari *database* yang dibuat. Contoh basis data untuk penyimpanan data stok minuman yang ada di salah satu warung di Universitas Pertamina dapat dilihat pada Tabel 3. Rancangan basis data lengkap dapat dilihat pada Lampiran C.



Gambar 3. Rancangan skema basis data *e-kantin* Universitas Pertamina

Tabel 3. *Stock* barang yang tersedia

No	Barcode	Name	Category	Unit	Price	Stock	Image
1	A001	Dalgona Coffee	Minuman	Buah	10000	20	
2	A002	ThaiTea	Minuman	Buah	7000	20	
3	A003	Ice Milo	Minuman	Buah	7000	20	

Secara paralel dengan perancangan basis data, rancangan halaman antarmuka sistem juga dibuat. Setelah kedua rancangan tersebut siap, dilakukan integrasi dimana antarmuka sistem dapat melakukan operasi sesuai yang direncanakan pada *use case* pada penjabaran sebelumnya. Sistem *e-kantin* memiliki beberapa halaman antarmuka yang dibuat yaitu: halaman login sistem, halaman utama, halaman profil, halaman produk, halaman supplier, halaman customer, halaman user, halaman penjualan, halaman stok masuk, halaman stok keluar, halaman laporan penjualan. Tampilan masing-masing halaman dapat dilihat pada Lampiran D.

Pada halaman *login*, pengguna dapat disaring sebelum masuk ke dalam sistem Kantin Universitas Pertamina, sistem login ini dilengkapi dengan adanya *username* dan *password* untuk admin (pemilik kantin) dan juga kasir di kantin tersebut. Adanya login ini dibuat untuk keamanan sistem pembayaran. Halaman utama (*dashboard*) akan muncul ketika admin atau pengguna sistem berhasil *login* sistem. Pada *dashboard* terdapat pilihan layanan yang telah disediakan seperti ikon *items*, *suppliers*, *customer*, dan *users*. Pada *dashboard* juga ditampilkan grafik pendapatan penjualan produk setiap harinya. Halaman profil menampilkan akun pengguna sistem yang sedang *login* di dalam sistem, pengguna (*admin*) dapat mengubah tampilan profil sesuai kemauan pengguna.

Pada halaman produk terdapat pilihan produk yang dijual oleh pengguna sistem *e-kantin* tersebut. Halaman produk menampilkan 3 pilihan layanan yaitu *categories*, *units*, dan *items*. Pada *categories* akan menampilkan tabel berupa jenis produk yang dijual seperti pada sistem ini yaitu makanan dan minuman, kemudian terdapat tampilan *units* yaitu satuan produk yang dijual misalnya dalam satuan buah atau kg, kemudian yang terakhir tampilan *items* yaitu tampilan produk yang dijual serta terdapat *barcode* suatu produk, antar produk tidak boleh memiliki *barcode* yang sama, serta pada tampilan *items* terdapat tabel *stock* produk yang tersedia.

Halaman *supplier* digunakan untuk menampilkan *supplier* dari pengguna sistem. Halaman ini digunakan untuk melihat, menghapus, ataupun mengubah data *supplier* yang telah ada, serta pada halaman ini dapat menambah *supplier* yang ada. Halaman *customer* menampilkan pembayaran di kantin pengguna. Halaman ini digunakan untuk melihat, menghapus, ataupun mengubah data *customer* yang telah ada, serta pada halaman ini dapat menambah *customer* yang ada. Halaman *user* menampilkan siapa saja yang menggunakan sistem pembayaran tersebut, biasanya terdiri dari admin dan kasir kantin. Halaman ini digunakan untuk melakukan pengelolaan terhadap *user*. Data *users* hanya dapat diubah oleh *admin* saja, jadi ketika kasir yang melakukan proses login maka tidak ada tampilan *users* pada *dashboard*. Pada tampilan *user* terdapat tabel yang

didalamnya mengandung *username*, *name*, *address*, dan level pengguna (*admin* atau kasir). Isi dari tabel-tabel tersebut dapat dihapus atau diubah.

Halaman *sales* seperti dicontohkan pada Gambar 4, menampilkan adanya proses transaksi penjualan suatu produk. Pada pilihan kasir akan terisi dengan sendirinya sesuai dengan pengguna yang *login* pada sistem, pada pilihan *customer* bisa menggunakan data *customer* yang telah dibuat atau menggunakan pilihan umum, pengisian *barcode* sesuai produk yang akan dibeli *customer* beserta jumlah produk yang akan dibeli. Setelah semua terisi maka klik *process payment* dan pembayaran berhasil, setelah pembayaran berhasil tampilan *invoice* akan kembali ke angka 0 kembali. Halaman stok masuk menampilkan banyaknya jumlah barang yang masih tersedia. Halaman *stock in* berada di ikon *transaction*, pada halaman ini menjabarkan pencatatan adanya jumlah produk yang masuk dari *supplier*. Halaman stok keluar menampilkan berapa banyak barang yang sudah keluar akibat adanya proses transaksi penjualan kepada *customer*, atau persediaan berkurang karena produk rusak atau kadaluarsa. Halaman laporan penjualan menampilkan adanya grafik transaksi penjualan milik pengguna sistem *e-kantin* Universitas Pertamina. Pada halaman *report* akan memberitahu berapa banyak jumlah produk yang dapat dijual setiap harinya.

Gambar 4. Halaman penjualan produk (*sales*)

Setelah perancangan sistem proses selanjutnya adalah tahap pengujian sistem. Pengujian merupakan proses verifikasi dan validasi. Verifikasi merupakan proses pengecekan kode pemrograman, apakah sudah secara tepat dapat mencerminkan model konseptual. Pada tahap *testing* ini juga dilakukan perbaikan pada *syntax error* yang terjadi pada pemrograman. Validasi merupakan proses untuk menentukan apakah model konseptual yang telah dirancang sudah mencerminkan sistem nyata, dalam hal ini validasi merupakan kegiatan uji coba terhadap sistem yang telah dibangun. Setelah melakukan *testing* didapatkan bahwa sistem telah memenuhi *requirement* yang dibuat sebelumnya. Sebelum masuk ke dalam sistem, aktor baik *admin* dan kasir harus melakukan proses *login*. Pada saat *login* hanya *admin* yang dapat mengelola informasi mengenai *users*. Pada menu *supplier*, *customer*, dan *products* sudah dapat dilakukan perintah tambah, hapus, dan *detail*. Menu *transaction* juga sudah dapat digunakan dengan baik sehingga jika terjadi pembelian barang pada menu *products* akan berkurang. Pada sistem ini sudah cukup baik untuk mempresentasikan kebutuhan dan keinginan *user*.

Sistem pembayaran berbasis *web* yang telah dibuat sudah dapat dijalankan atau dioperasikan untuk setiap penggunanya, jika dalam kasus ini sistem sudah dapat digunakan oleh penjual di Kantin Universitas Pertamina dengan tampilan *dashboard*, *supplier*, *customer*, *products*, *transaction*, dan *report*. Sistem yang telah dibangun memerlukan pemeliharaan sehingga dapat meminimalkan *error* yang mungkin terjadi. Pengembangan sistem dari segi tampilan dan fungsinya perlu terus dilakukan dengan mengikuti perkembangan zaman dan kebutuhan dari *user*.

Kesimpulan

Berdasarkan hasil pembahasan yang telah dijelaskan maka dapat disimpulkan bahwa untuk menyelesaikan permasalahan antrian yang terjadi di kantin Universitas Pertamina diperlukan pengembangan sistem pembayaran berbasis *web*. Sistem pembayaran berbasis *web* di kantin Universitas Pertamina dibangun dengan menggunakan MySQL untuk basis data, phpMyAdmin untuk mengelola database MySQL, javascript digunakan sebagai bahasa pemrograman, dan XAMPP versi 7.4.4 yang digunakan sebagai pembuatan *web server* secara lokal yang dapat melayani tampilan halaman *web*. Terdapat 2 aktor yang dapat mengakses aplikasi ini yaitu *admin* yang merupakan pemilik toko dan kasir sesuai dengan *use case specification*. Sistem pembayaran berbasis *web* yang telah dirancang memiliki beberapa fitur yaitu *supplier*, *customer*, *products*, *transactions*, dan *report*.

Pada menu *supplier*, *customer*, dan *products* sudah dapat dilakukan perintah tambah, hapus, dan *detail*. Menu *transaction* juga sudah dapat digunakan dengan baik sehingga jika terjadi pembelian barang pada menu *products* akan berkurang secara *real time*. Sistem ini sudah cukup baik untuk mempresentasikan kebutuhan dan keinginan *user*. Penggunaan sistem pembayaran berbasis *web* ini dinilai dapat mengurangi tingkat antrian pembelian makanan dan minuman di Kantin Universitas Pertamina, hal ini dapat dilihat pada toko yang telah menggunakan sistem ini lebih dulu. Berdasarkan observasi yang dilakukan, antrian toko yang belum menggunakan sistem pembayaran *online* lebih banyak dibandingkan toko yang sudah mulai menggunakan sistem pembayaran *online*.

Pesatnya perkembangan teknologi saat ini menyebabkan sistem *e-kantin* yang dirancang dapat dikembangkan lebih lanjut. Pengembangan lebih lanjut dari sistem *e-kantin* ini dapat berupa aplikasi *mobile* sehingga akan lebih menarik dari segi visualnya, kemudian pengembangan lebih lanjut dapat dilakukan terintegrasi dengan sistem pembayaran *e-wallet* seperti Gopay, OVO, Link aja, dan sebagainya. Pengujian *user acceptance test* juga dapat menjadi pertimbangan untuk pengembangan sistem selanjutnya sehingga sistem yang dibangun akan lebih sesuai dengan keinginan *user* serta dapat mencapai *user satisfaction*.

Daftar Pustaka

- [1] Universitas Pertamina, "Sejarah – Universitas Pertamina," *universitaspertamina.ac.id*, 2020. <https://universitaspertamina.ac.id/sejarah/> (accessed Nov. 26, 2020).
- [2] S. Nurhayati, "Perancangan Sistem Informasi Pembayaran Elektronik Pada Kantin XYZ," *Komputika J. Sist. Komput.*, vol. 8, no. 1, pp. 25–29, 2019.
- [3] A. Budirahma, T. Rizal, and H. Vidyanyingtyas, "Aplikasi Sistem Informasi Pengelolaan Transaksi Kantin Fakultas Teknik Universitas Telkom Berbasis Android," *eProceedings Eng.*, vol. 1, no. 1, pp. 462–469, 2014.
- [4] A. Rochman, A. Sidik, and N. Nazahah, "Perancangan Sistem Informasi

- Administrasi Pembayaran SPP Siswa Berbasis Web di SMK Al-Amanah,” *J. Sisfotek Glob.*, vol. 8, no. 1, 2018.
- [5] K. R. T. dan P. T. KEMENRISTEK, “SRV4 PDDIKTI□: Pangkalan Data Pendidikan Tinggi,” *forlap.ristekdikti.go.id*, 2020. <https://forlap.ristekdikti.go.id/perguruantinggi/detail/RkM0OTE1OUUtQTVVDNy00QkUyLUIxRjktNTg1RDQwNzNFODhC> (accessed Nov. 26, 2020).
- [6] A. Fatah Rusdi, “E-Kantin Unikom Sebagai Layanan Pemesanan Berbasis Web,” Universitas Komputer Indonesia, 2018.
- [7] E. Astriyani, M. M. Sari, and H. Herman, “Perancangan Sistem Informasi Pembayaran Spp Berbasis Web Menggunakan Notifikasi SMS Gateway (Studi Kasus: SMP Puspita Tangerang),” *J. Cerita*, vol. 6, no. 1, pp. 106–116, 2020.
- [8] W. Abdillah and J. Hartono, *Partial Least Square (PLS): alternatif structural equation modeling (SEM) dalam penelitian bisnis*, 1st ed., vol. 22. Yogyakarta, 2015.
- [9] H. M. Jogiyanto, *Analisis dan Desain (Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis)*, 1st ed. Yogyakarta: Penerbit Andi, 2017.
- [10] A. Oktaviani, D. Sarkawi, and A. Priadi, “Perancangan Aplikasi Penjualan Dengan Metode Waterfall Pada Koperasi Karyawan Rsud Pasar Rebo,” *PETIR J. Pengkaj. dan Penerapan Tek. Inform.*, vol. 11, no. 1, pp. 9–24, 2018.
- [11] G. W. Sasmito, “Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal,” *J. Inform. J. Pengemb. IT*, vol. 2, no. 1, pp. 6–12, 2017.
- [12] S. T. Safitri and D. Supriyadi, “Rancang Bangun Sistem Informasi Praktek Kerja Lapangan Berbasis Web dengan Metode Waterfall,” *J. Infotel*, vol. 7, no. 1, pp. 69–74, 2015.
- [13] P. P. Widodo and H. Prabowo, “Menggunakan Uml,” *Bandung Inform.*, vol. 19, pp. 393–403, 2011.

Lampiran

Lampiran A. Tabel Use Case Specification

Tabel A.1. RF-1: *Login*

Nama Use Case	<i>Login</i>	
Aktor	Pemilik tempat makan, kasir	
Entry Condition	Aktor membuka halaman <i>login</i>	
Scenario	Aktor	Sistem
	1. Input <i>E-mail</i> dan <i>password</i> 2. Klik tombol <i>login</i>	3. Menampilkan menu utama
Exit Condition	Sistem menampilkan halaman utama menu	
Scenario Alternatif	Aktor	Sistem
	1. Input <i>E-mail</i> dan <i>password</i> 2. Klik tombol <i>login</i>	3. Menampilkan notifikasi gagal <i>login</i>
Exit Condition	Sistem menampilkan halaman login kembali dan pesan gagal <i>login</i>	

Tabel A.2. RF-2: Pilihan menu

Nama Use Case	Pilihan menu	
Aktor	Pemilik tempat makan, kasir	
Entry Condition	Aktor membuka halaman <i>login</i>	
Scenario	Aktor	Sistem
	1. Klik pilihan restaurant 2. Kik pilihan makanan	3. Menampilkan menu makanan
Exit Condition	Sistem menampilkan menu makanan	

Tabel A.3. RF-3: *Check out*

Nama Use Case	<i>Check out</i>	
Aktor	Kasir	
Entry Condition	Aktor membuka halaman <i>Check Out</i>	
Scenario	Aktor	Sistem
	1. Klik tombol <i>check out</i> 2. Klik tombol pembayaran	3. Menampilkan pesanan makanan dan harga total makanan
Exit Condition	Sistem menampilkan halaman pembayaran	

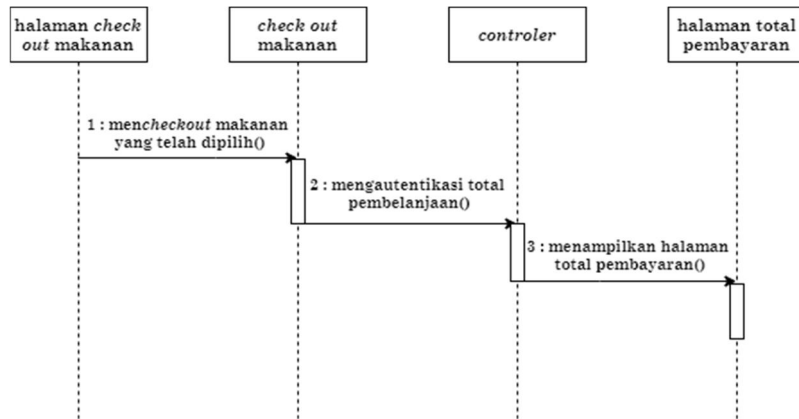
Tabel A.4. RF-4: Cetak resi

Nama Use Case	Cetak Resi	
Aktor	Kasir	
Entry Condition	Aktor membuka halaman <i>Check Out</i>	
Scenario	Aktor	Sistem
	2. Klik tombol cetak resi pada halaman pembayaran jika telah selesai melakukan pembayaran	3. Sistem mengunduh resi pembayaran
Exit Condition	Aktor mengunduh resi untuk kemudian diberikan kepada <i>customer</i>	

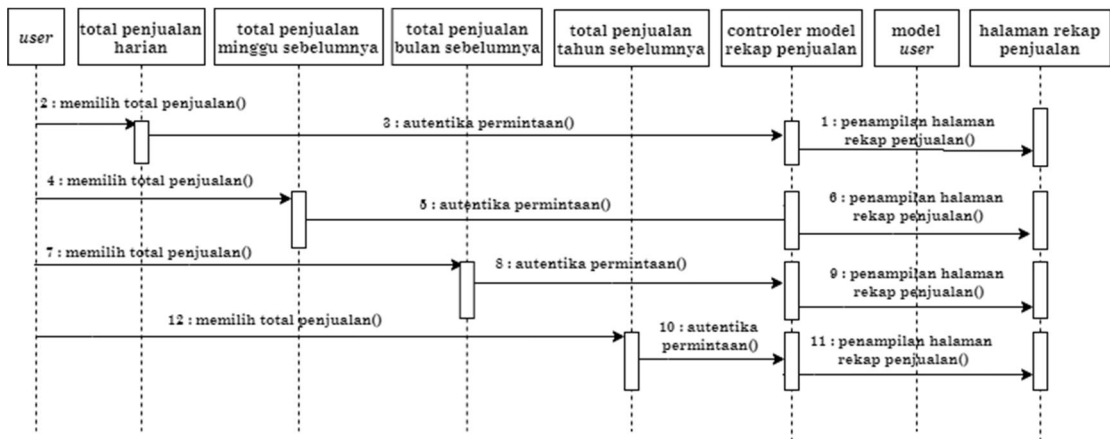
Tabel A.5. RF-6: *Logout*

Nama Use Case	<i>Logout</i>	
Aktor	Pemilik tempat makan, kasir	
Entry Condition	Aktor telah berada didalam menu aplikasi	
Scenario	Aktor	Sistem
	1. Aktor mengklik tombol <i>logout</i>	2. sistem menampilkan halaman <i>logout</i>
Exit Condition	Aktor telah <i>logout</i> dan sistem kembali menampilkan halaman <i>login</i>	

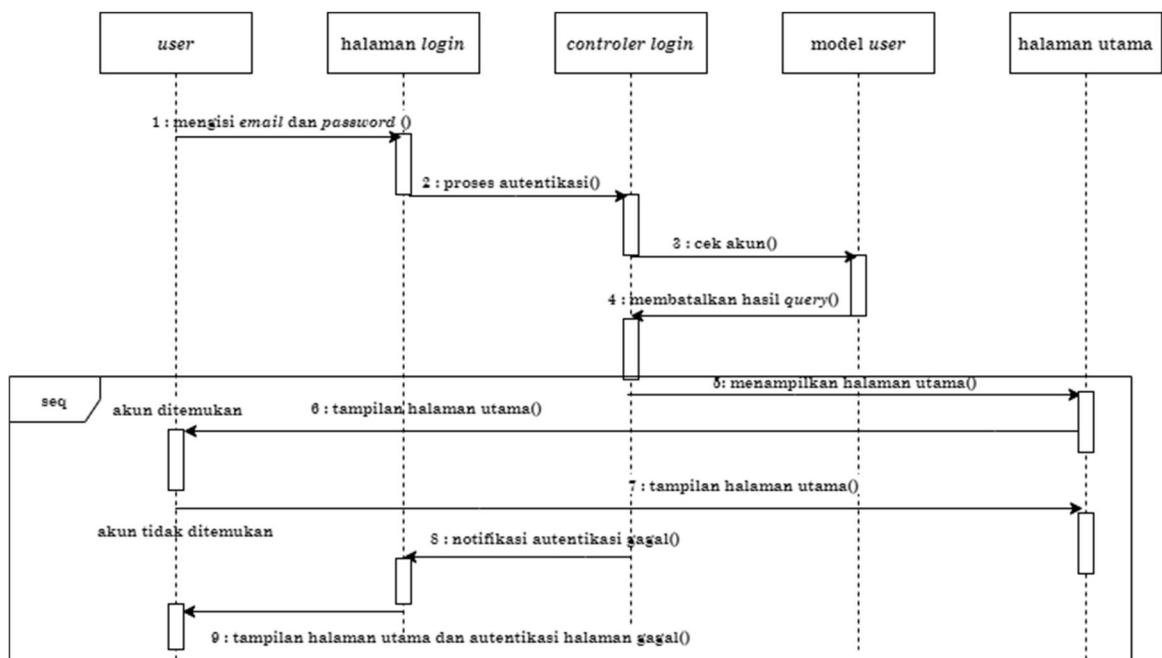
Lampiran B. Sequence Diagram



Gambar B.1. Sequences diagram halaman login



Gambar B.2. Sequences diagram halaman pemesanan makanan



Gambar B.3. Sequences diagram halaman rekap hasil penjualan

Lampiran C. Contoh tabel dalam basis data**Tabel C.1.** Kebutuhan data *supplier*

No	Name	Phone	Adress	Description
1	Toko A	85959941555	Rawa Simprug, Jakarta Selatan	Supplier Kopi
2	Toko B	8122033440	Kebon Nanas I, Jakarta Selatan	Supplier Gula
3	Toko C	85976548777	Bandung	Semua ada disini




Tabel C.2. Kebutuhan data pemilik kantin

No	Name	Gender	Phone	Address
1	Meisya Indriani	F	865743657	Brebes
2	Disa Maya	F	876543688	Nganjuk

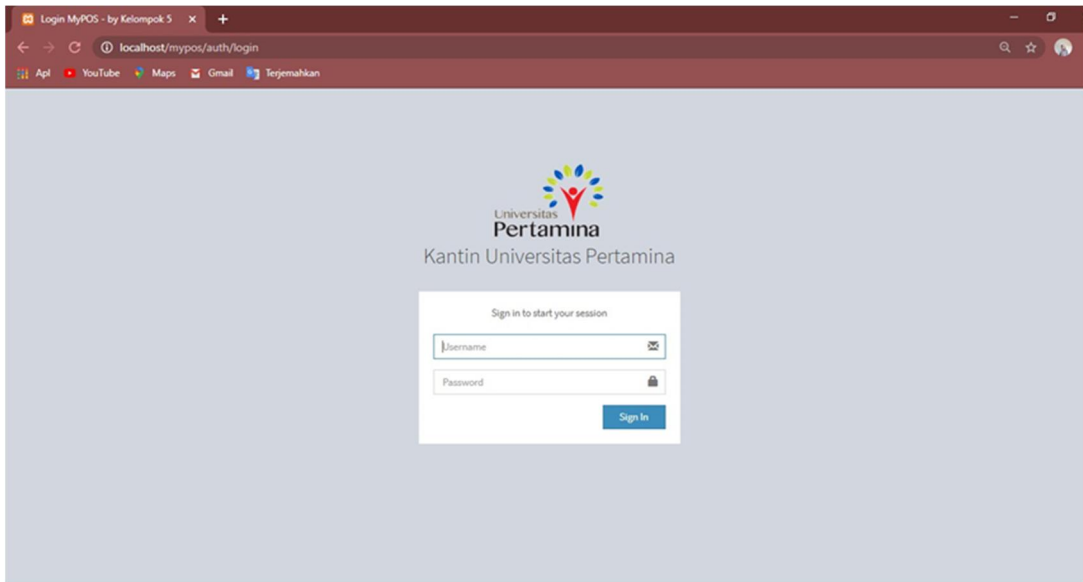
Tabel C.3. Jumlah *order* yang terjual

Product Item	Qty	Date
Thai Green Tea	20	5/4/2020
Thai Tea Original	20	5/4/2020
Ice Milo	10	5/4/2020

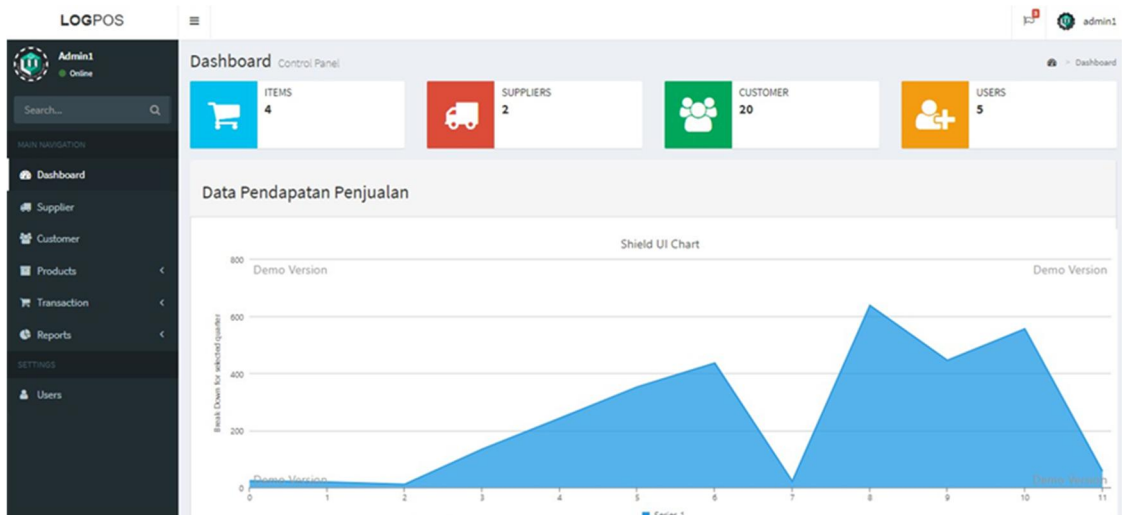
Tabel C.4. *Stock* minuman yang tersedia

No	Barcode	Name	Category	Unit	Price	Stock	Image
1	A001	Dalgona Coffee	Minuman	Buah	10000	20	
2	A002	ThaiTea	Minuman	Buah	7000	20	
3	A003	Ice Milo	Minuman	Buah	7000	20	

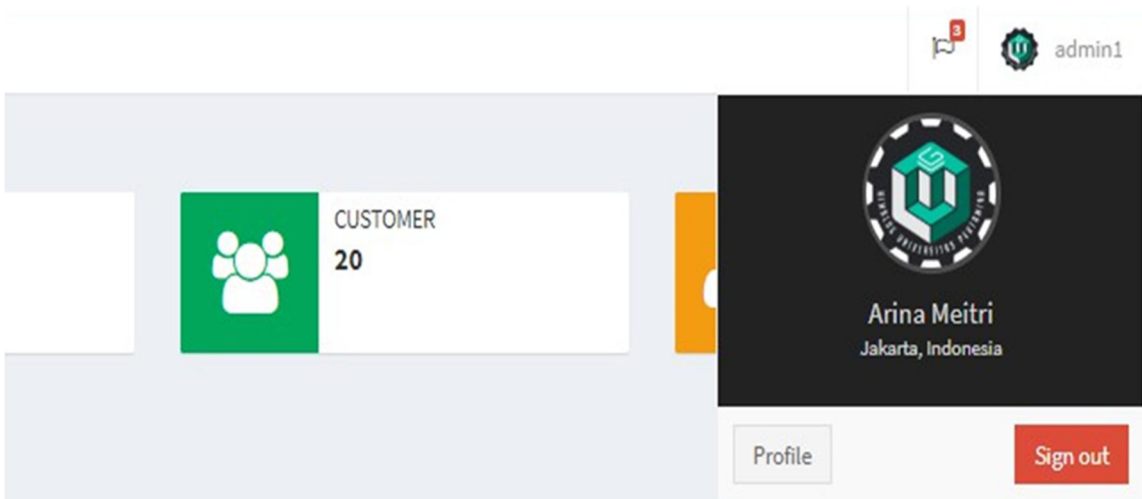
Lampiran D. *Desain* antarmuka sistem



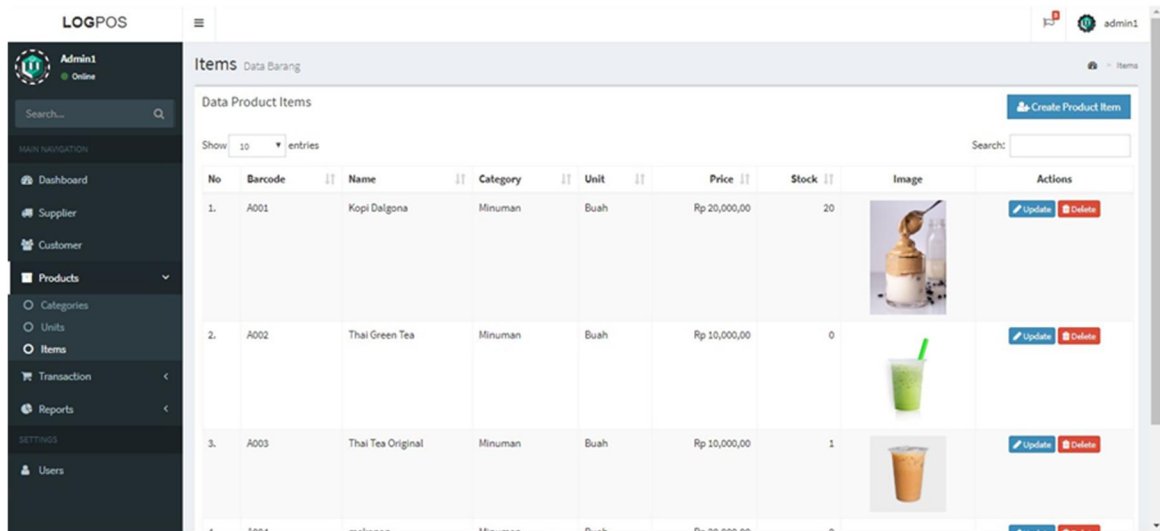
Gambar D.1. Halaman *login* sistem



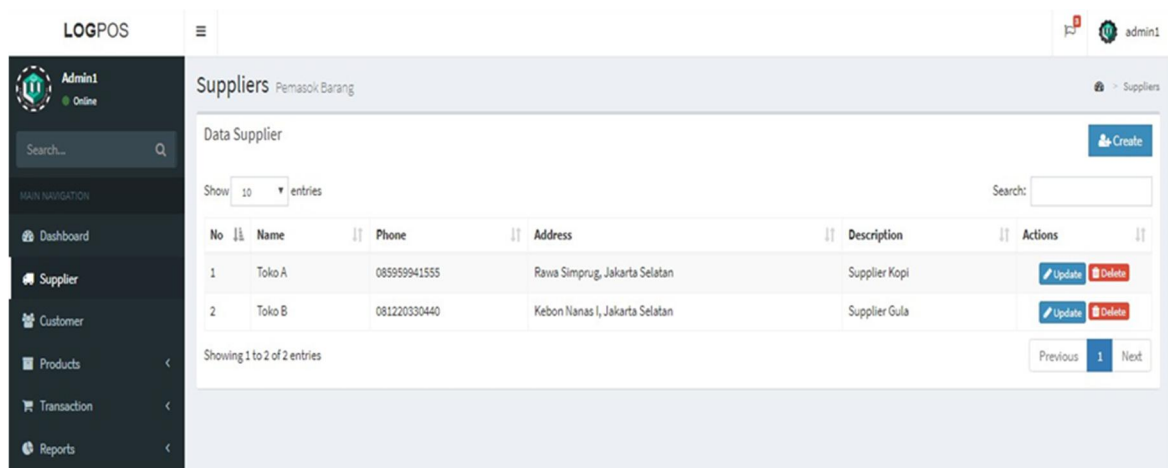
Gambar D.2. Halaman utama (*dashboard*)



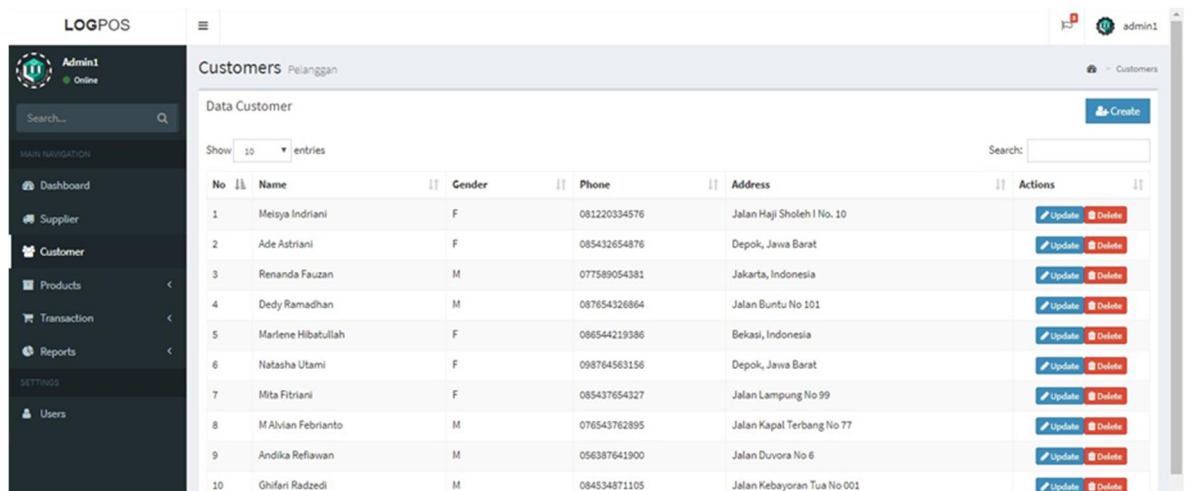
Gambar D.3. Halaman profil pengguna sistem



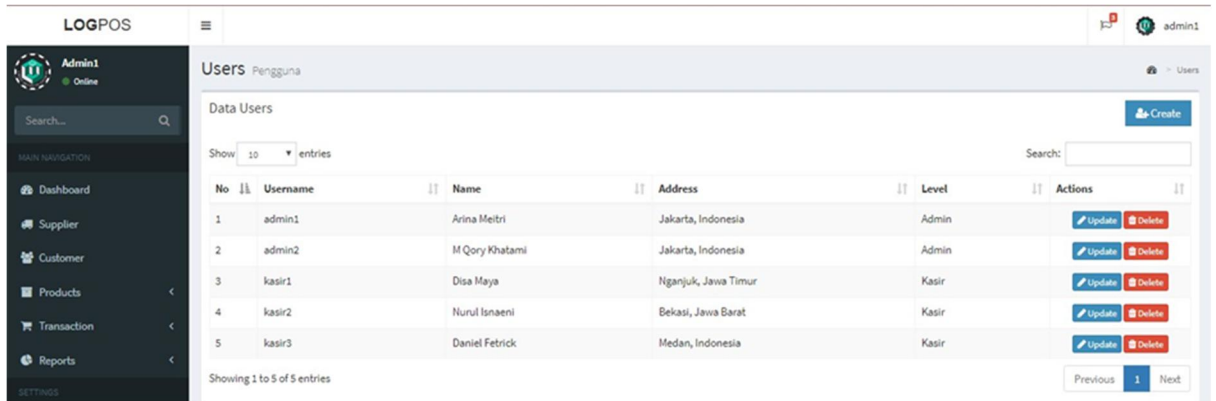
Gambar D.4 Halaman *items*



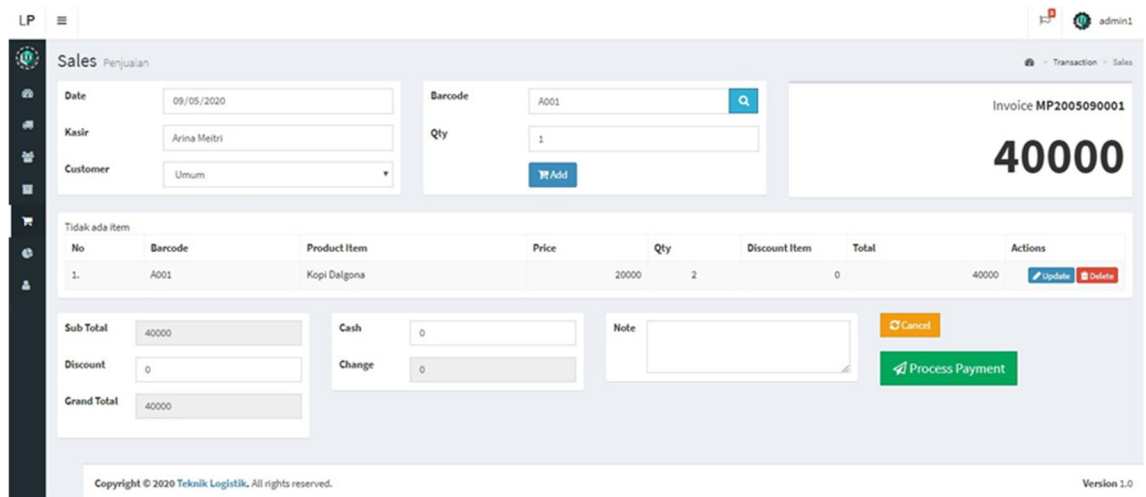
Gambar D.5. Halaman *supplier*



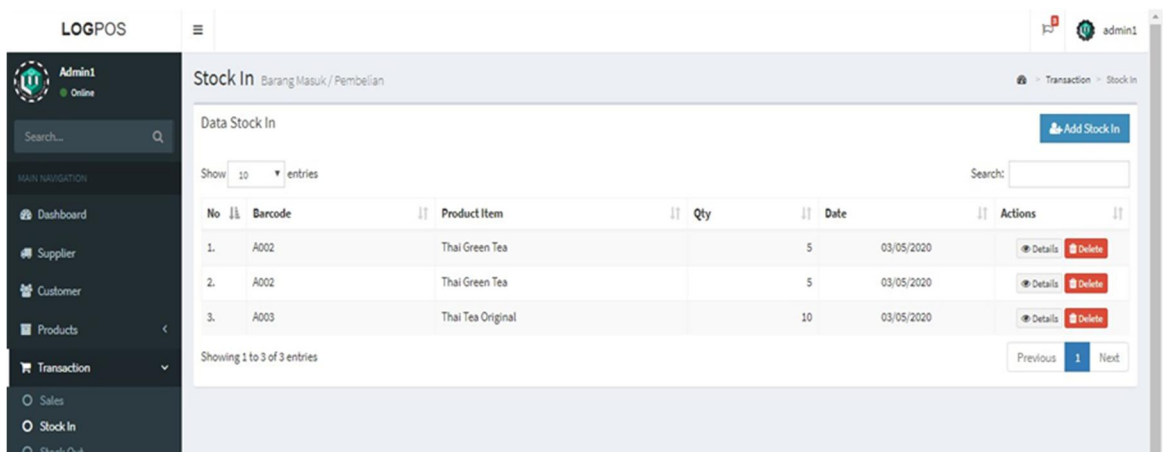
Gambar D.6. Halaman *customers*



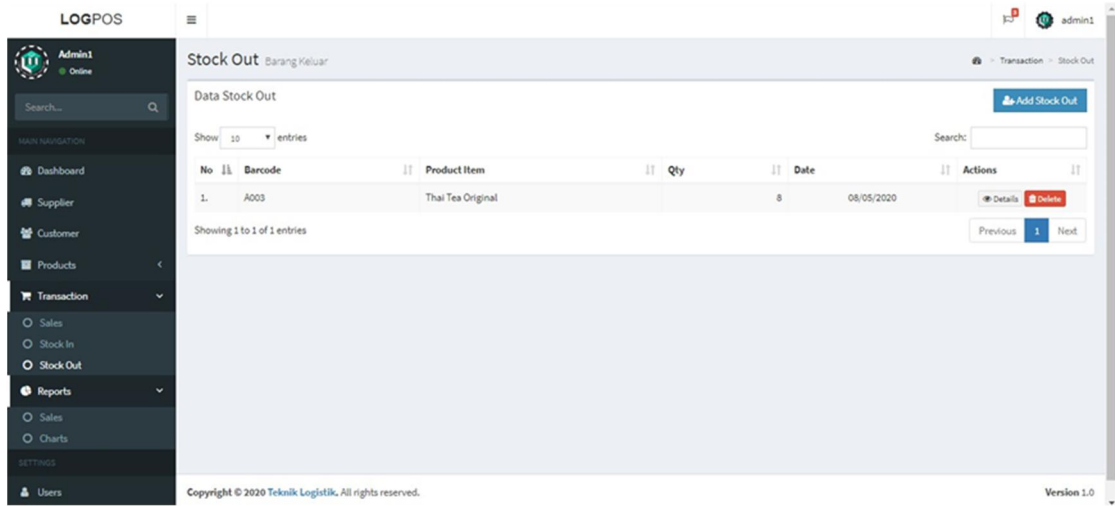
Gambar D.7. Halaman users



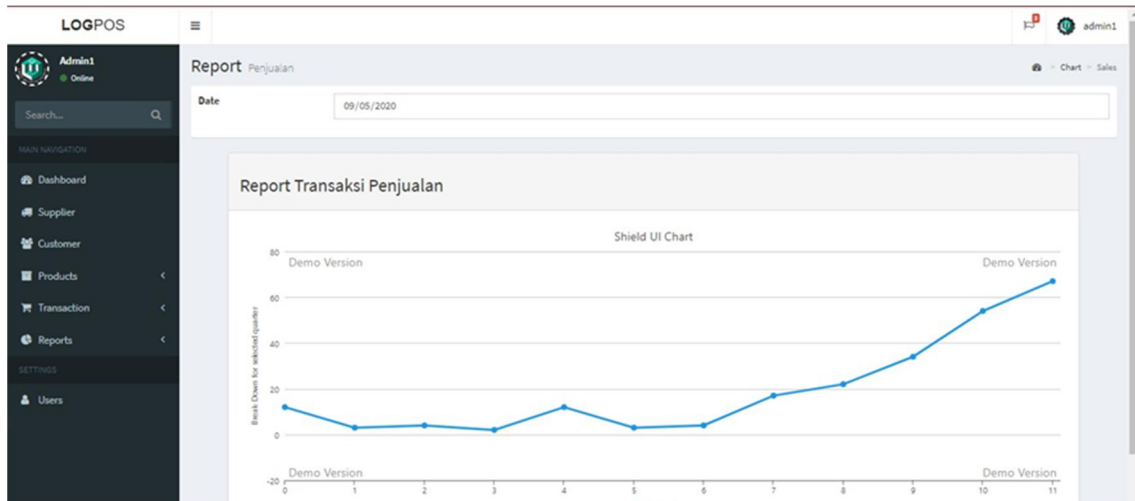
Gambar D.8. Halaman penjualan produk (sales)



Gambar D.9. Halaman stock



Gambar D.10. Halaman transaksi *stock out*



Gambar D.11. Halaman *report* penjualan *items*